

readme.md for @push.rocks/smartcontext

A zero-dependency module for hierarchical async context management in Node.js, built on `AsyncLocalStorage`.

Install

```
npm install @push.rocks/smartcontext  
# or  
pnpm install @push.rocks/smartcontext
```

Usage

`@push.rocks/smartcontext` provides scoped, hierarchical key-value stores that follow async execution flow. Child scopes inherit parent data, can add or delete keys without affecting the parent, and automatically clean up when the scope exits.

This is useful for log enrichment, request-scoped metadata, transaction tracking, and any scenario where contextual data needs to flow through deeply nested async calls.

Basic Setup

```
import { AsyncContext } from '@push.rocks/smartcontext';  
  
const ctx = new AsyncContext();  
  
// Add data to the root store  
ctx.store.add('userId', 'u_123');  
console.log(ctx.store.get('userId')); // 'u_123'
```

Scoped Execution with `runScoped`

`runScoped` creates an isolated child store for the duration of the callback. Inside the callback, `ctx.store` transparently points to the child store. The child inherits all parent data but any additions or deletions are local to the child.

```
ctx.store.add('requestId', 'req_abc');

await ctx.runScoped(async () => {
  // Child store sees parent data
  console.log(ctx.store.get('requestId')); // 'req_abc'

  // Add child-only data
  ctx.store.add('spanId', 'span_001');
  console.log(ctx.store.get('spanId')); // 'span_001'
});

// After scope exits, child data is gone
console.log(ctx.store.get('spanId')); // undefined
// Parent data is untouched
console.log(ctx.store.get('requestId')); // 'req_abc'
```

Deleting Keys in a Child Scope

Deleting a parent key inside a child scope only shadows it within that scope. The parent retains the original value.

```
ctx.store.add('token', 'secret_value');

await ctx.runScoped(async () => {
  ctx.store.delete('token');
  console.log(ctx.store.get('token')); // undefined (shadowed)
});

console.log(ctx.store.get('token')); // 'secret_value' (parent unaffected)
```

Parallel Scopes

Multiple concurrent `runScoped` calls each get their own isolated child store, preventing data collisions across async tasks.

```
await Promise.all([
  ctx.runScoped(async () => {
    ctx.store.add('worker', 'A');
    // Only this scope sees worker=A
  }),
  ctx.runScoped(async () => {
    ctx.store.add('worker', 'B');
    // Only this scope sees worker=B
  }),
]);

console.log(ctx.store.get('worker')); // undefined
```

Retrieving All Data

`store.getAll()` returns a merged view of the store hierarchy, with child values overriding parent values and deleted keys excluded.

```
ctx.store.add('env', 'production');
ctx.store.add('region', 'eu-west-1');

await ctx.runScoped(async () => {
  ctx.store.add('traceId', 'tr_xyz');
  console.log(ctx.store.getAll());
  // { env: 'production', region: 'eu-west-1', traceId: 'tr_xyz' }
});
```

API Reference

AsyncContext

Method / Property	Description
<code>store</code>	The current <code>AsyncStore</code> (root or scoped child)
<code>runScoped(fn)</code>	Execute <code>fn</code> with an isolated child store

AsyncStore

Method	Description
<code>add(key, value)</code>	Set a key-value pair
<code>get(key)</code>	Get a value (checks local store, then parent chain)
<code>delete(key)</code>	Remove a key (shadows parent key if inherited)
<code>getAll()</code>	Get all key-value pairs merged from the full parent chain

License and Legal Information

This repository is under the [MIT License](#). Please note that the MIT License does not grant permission to use the trade names, trademarks, service marks, or product names of the project, except as necessary for reasonable use in describing the origin of the work.

Trademarks

This project is owned and maintained by Task Venture Capital GmbH. The names and logos associated with Task Venture Capital GmbH are trademarks of Task Venture Capital GmbH and are not included within the scope of the MIT license granted herein. Usage must be approved in writing by Task Venture Capital GmbH.

Company Information

Task Venture Capital GmbH Registered at District Court Bremen HRB 35230 HB, Germany

For any legal inquiries, please contact us at hello@task.vc. By using this repository, you acknowledge that you have read this section and agree to comply with its terms.

Revision #3

Created 2026-03-28 11:09:53 UTC by foss.global Team

Updated 2026-03-28 12:16:30 UTC by foss.global Team